NTNU
Kunnskap for en bedre verden

Institutt for datateknikk og informasjonsvitenskap (IDI)

Eksamensoppgave i TDT4300 Datavarehus og datagruverdrift - Vår 2014 (Sensurveiledning)

Faglig kontakt under eksamen: Heri Ramampiaro

Tlf.: 99027656

Eksamensdato: 03. juni 2014

Eksamenstid (fra-til): 09:00 - 13:00

Hjelpemiddelkode/Tillatte hjelpemidler: D – Ingen trykte eller håndskrevne tillatt. Kun typegodkjent kalkulator er tillatt

Annen informasjon: Svar **kort** og **konsist** på alle spørsmålene. Stikkord foretrekkes fremfor lange forklaringer.

Målform/språk: Bokmål

Antall sider: 5

Antall sider vedlegg: 0

Kontrollert av:

Dato Sign

Oppgave 1 (15%):

1. Forklar hva er datavarehus er.
2. Beskriv de viktigste forskjellene mellom et datavarehus og et operasjonelt databasesystem.
3. Hvilke prosesser inngår typisk i det som forkortes ETL.
4. Beskriv hva en datakube er.
5. Hvilke data representerer en 0-D cuboid (også kalt apex cuboid)?

Oppgave 2 (25%):

Du er i et prosjekt og skal lage et datavarehus med karakterdata som skal brukes for å undersøke hvordan ulike institusjoner og studieprogram benytter karakterskalaen. Dere skal kun støtte høyere utdanning hvor alle bruker samme karakterskala.

Statistikk som skal genereres er typisk karakterfordelingen (andel % for hver karakter), snittkarakter og lignende. Dere ønsker å undersøke i forhold til forskjellige grupper av studenter (alder, kjønn og annen informasjon som er relatert til person). Alle høyere utdanningsinstitusjoner i Norge (universiteter og høgskoler) tilbyr forskjellige studieprogram som kan kategoriseres med fagdisiplin (økonomi, samfunnsvitenskap, medisin, psykologi etc) og med nivå (bachelor, master, phd). Et studieprogram består bestandig av en samling emner og studentene får karakterer i enkeltemner. Andre aspekter som kan være av interesser er karakterfordeling over tid og karakterfordeling i forhold til landsdeler og byer.

Beskrivelsen er litt upresist formulert og det er en del av oppgaven å velge ut hva som skal være med. Vi er først og fremst ute etter at du skal vise modelleringsprinsippet for datavarehus. Forklar kort eventuelle forutsetninger du finner det nødvendig å gjøre.

1. Lag et snøflak eller stjerneskjema for dette caset og beskriv hva som er forskjellen mellom disse to skjematypene.
2. Lag to forskjellige konsepthierarkier. Beskriv forskjellen mellom et hierarkisk og gitter-basert konsepthierarki (hierarchical vs. lattice).
3. Lag en tabell som eksemplifiserer hvilke data som kan genereres fra datavarehuset. Legg vekt på å få fram aggregeringsprinsippet.

Oppgave 3 (15%):

1. Forklar hva som hovedforskjellene mellom datavarehus (data warehouse) og datagruvedrift (data mining).

Svar:
Datavarehus: Analyser, rapporter og spørringer.
Delvis å hente ut forhåndsdefinert "***kjent informasjon" (men ikke kjente resultater***)
Datagruvedrift: ***oppdage ny informasjon (knowledge discovery)***

2. Forklar hvorfor assosiasjonsregler ikke kan regnes som en prediksjon (prediction) mens klassifisering er det.

Svar: AR kan ikke regnes som prediksjon fordi den bygger regler basert på eksisterende data og har ingen kausalitet. Klassifisering er derimot prediskjon da man prøver å "spå" klassetilhørighet basert på en nåværende kunnskap (feks. trenet klassifiseringsmodell).

3. Hva er hovedforskjellene mellom klassifisering og klynging (clustering)? Forklar.

Svar: Her er det nok at studentene fokuserer forklaringene på "supervised learning" (klassifisering) vs. "unsupervised learning" (clustering).

4. Forklar hovedfordelene med flat klynging (flat clustering) som feks. K-means mot hierarkisk klyning (hierarchical clustering) og omvendt (d.v.s hierarkisk mot flat).

Svar: Her er kreves at studentene får med seg hvilke aspekter relater til hastighet, valg av sentroider, sensitivitet for støy og outliers, osv.

Oppgave 4 (20%):

Amazon Inc. er kjent for å analysere hva kundene deres kjøper. De er opptatt av å ha balanse mellom etterspørsel av varene og lagerbeholdningen. Defor bestemmer de seg for å bruke assosiasjonsregelanalyse og -mining til dette formålet.

Før de setter igang bestemmer de seg for å gjøre noen enkle lavskala-analyser på noen få varer som kundene har kjøpt sammen i det siste:

```
t1: {Headphones, iPod}

t2: {Headphones,  iPad, iTunes-giftcard, Running shoes}

t3: {iPod, iPad, iTunes-giftcard, Bike-computer}

t4: {Headphones, iPod, iPad, iTunes-giftcard}

t5: {Headphones, iPod, iPad, Bike-computer}
```

1. Hva menes med **støtteantall** (support count) og **støtte** (support)? Hvor mye blir støtteantall og støtte for `{Headphones, iPod}`?

Svar: Støtteantall: Frekvens av forekomst av et elementsett. Støtte: Andel av transaksjoner som inneholder et elementsett. sc(`{Headphones, iPod})= 3, s({Headphones, iPod})= 3/5 = 60%.`

2. Anta at minimum support *minsup = 0.5*. Hvilke elementsett er **frekvente elementsett**?

Svar: munsup=0.5 betyr min sup.count på 3.

Disse oppyller dette kravet på minsup=0.5

1-sett elementsett: {Headphones}, {iPod}, {iPad}, {iTunes-giftcard}

2-sett elementsett: {Headphones, iPod}, {Headphones, iPad}, {iPod, iPad}, {iPad, iTunes-giftcard}

3-sett elementsett: ingen oppfyller minsup-kravet.

3. Anta regelen `{iPod, iPad} => iTunes-giftcard`

   a) Hva blir verdien av **støtte** (s) og **konfidens** (c) her?

Svar: Støtte, s=støtte({iPod, iPad, iTunes-giftcard})=2/5=0.4. Konfidens, c=støtte({{iPod, iPad, iTunes-giftcard}})/støtte({iPod, iPad})=2/3=0.67

   b) Hvorfor er "brute-force" regelgenereringsmetoden generelt er uegnet til assosiasjonsregeloppdagelser?

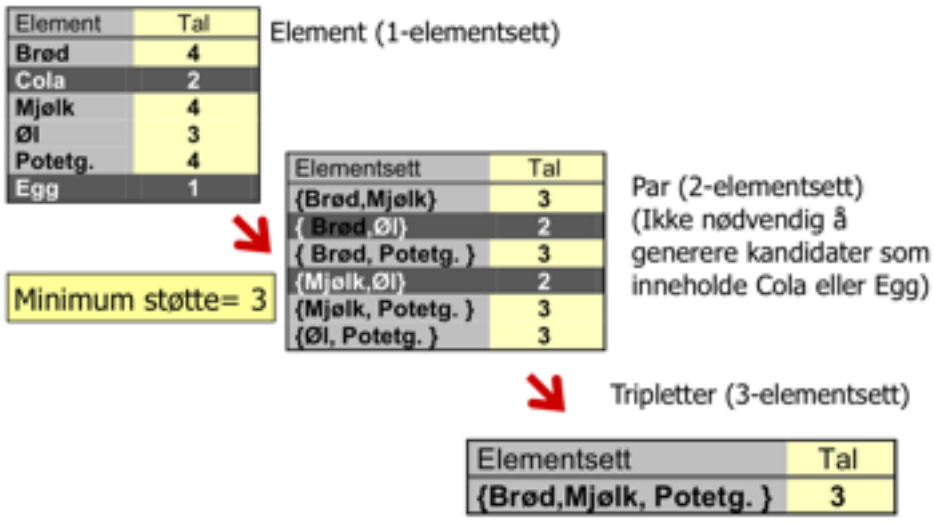
Svar: Hovedproblemet er tidskompleksitet siden en brute-force algoritmer ville se etter alle mulige kombinasjoner, også de som ikke er nødvendigvis interessante jobbe videre med.

   c) Hvordan kan "apriori-algoritmen" utnyttes til assosiasjonsregeloppdagelser? Tips: vi er ute etter apriori-prinsippet.

Svar: Her kreves forklaring på anti-monoton-egenskapene for å "prune" ikke frekvente elementsett.

   d) Forklar hvordan aprori-algoritmen er bygd opp ved å bruke eksempel-transaksjonene over som utgangspunkt. Anta fortsatt at *minsup=0.5*.

Svar: Her skal studentene lage noe tilsvarende dette:



4. Nevn *minst tre ting* som påvirker kompleksiteten på regelgenerering. Nevn deretter *minst en ting* man kan gjøre for å møte/løse utfordringene forbundet med denne kompleksiteten.

Svar:

- **Valg av minimum støtte (minsup)**
  - lavere terskel resulterer i flere frekvente elementsett
  - fører typisk til høyre antall på kandidater og maksstørrelse på frekvente elementsett
- **Dimensjonalitet** (antall av element) på datasettet
  - mer plass nødvendig for å lagre støtteantall (support count) til elementene
  - hvis antall på frekvente element også øker kan beregning og IO-kostnader også øke
- **Størrelse på databasen**

- Apriori gir flere pass gjennom databasen
  ➔ køyretid vil øke med antall på transaksjoner
- **Gjennomsnittlig transaksjonsbredde**
  - Økt transaksjonsbredde
    ➔ (typisk) økt maksstørrelse på frekvente elementsett

Oppgave 5 (25%):

Skandiabanken ASA vil finne ut om hvilke av deres kunder kommer til å kjøpe bil i fremtiden. Tabellen under viser informasjon om noen av kundene deres som allerede har kjøpt bil.

ID	Alder	Inntekt	Arbeidstype	Kredittverdighet	Bilkjøper
1	<= 30	Høy	Fulltid	Passe	Nei
2	<= 30	Høy	Fulltid	Høy	Nei
3	31 - 40	Høy	Fulltid	Passe	Ja
4	> 40	Middels-høy	Fulltid	Passe	Ja
5	> 40	Lav	Deltid	Passe	Ja
6	> 40	Lav	Deltid	Høy	Nei
7	31 - 40	Lav	Deltid	Høy	Ja
8	<= 30	Middels-høy	Fulltid	Passe	Nei
9	<= 30	Lav	Deltid	Passe	Ja
10	> 40	Middels-høy	Deltid	Passe	Ja
11	<= 30	Middels-høy	Deltid	Høy	Ja
12	31 - 40	Middels-høy	Fulltid	Høy	Ja
13	31 - 40	Høy	Deltid	Passe	Ja
14	> 40	Middels-høy	Fulltid	Høy	Nei

1. Bruk tabellen over og **Hunt's algoritmen** som utgangspunkt til å indusere et beslutningstre. Det forventes ikke at du skal lage et komplett tre. Det holder at du viser du har forstått prinsippet.

Svar: Her må studentene vise at de har forstått prinsippet bak Hunt's algoritmen. Jo mer detaljer de har med, desto mer poeng får de.

2. For å kunne finne ut om et splitt (split) er bra bruke en ofte å måle homogenitet av kandidatnoden for splittingen. Hvordan måles homogeniteten?

Svar: Homogenitet måles ved hvordan klassene er fordelt på en (kandidat)splittnode. En node som har lik fordeling på klassen er ikke homogen (50:50) vs. 100:00 (homogen).

3. Anta at du starter splittingen din på "Arbeidstype". Finn **GINI** og **Entropy**-verdiene for denne noden. Gitt GINI og Entropy som følgende:

$$GINI(t) = 1 - \sum_j p(j|t)$$

$$Entropy(t) = -\sum_j p(j|t) \log p(j|t)$$

Her er p(j |t) sannsynligheten for klassen j gitt noden t (d.v.s andelen av klassen j i noden t).

Svar: Ved noden t="Arbeidstype" er klassene fordelt på 5 av 14 "nei"-klasse og 9 av 14 "ja"-klasse. Dette betyr GINI (t) = 1 - ((5/14)^2 + (9/14)^2) =0.46, ENTROPY(t)=-((5/14)log(5/14) + (9/14)log(9/14))=0.28.

4. Gitt $GINI_{splitt}$ som følgende:

$$GINI_{split} = \sum_{i=0}^{k} \frac{n_i}{n} GINI(i)$$

Her er n_i antall elementer i node i og n total elementer i foreldernoden p. Bruk dette til å finne ut om "`Arbeidstype`" eller "`Kredittverdiget`" er best å starte splittingen med.

Svar:

For noden "Arbeidstype" kan vi splitte på "Fulltid" og "Deltid" som gir GINI(t) som følgende:

GINI(Fulltid) = 1 - ((4/7)^2 - (3/7)^2) = 1 - 0.327 - 0.187 = 0.486

GINI(Deltid) = 1 - ((1/7)^2 - (6/7)^2)= 1 - 0.02 - 0.735 = 0.245

GINI_splitt = 7/14*0.486 + 7/14*0.245 = 0.366

For noden "Kredittverdiget" kan vi gjøre tilsvarende beregning:

GINI(Passe) = 1 - ((2/8)^2 - (6/8)^2) = 1 - 0.0625 - 0.5625 = 0.375

GINI(Høy) = 1 - ((3/6)^2 - (3/6)^2)= 1 - 0.25 - 0.25 = 0.5

GINI_splitt = 8/14*0.375 + 6/14*0.5 = 0.429

"Kredittverdiget" har høyre GINI_splitt og er dermed dårligere valg enn "Arbeidstype".